2nd International Conference on Electrical Systems Design & Technologies, Hammamet Tunisia, Nov. 8-10, 2008

Evaluation of metaheuristics performances in manipulation of alternative routing

M. Souier
Laboratory of Automatic of Tlemcen, University of Abou bekr Belkaïd,PoBox 230, Tlemcen 13000, Algeria
m_souier@mail.univ-tlemcen.dz

A. Hassam
Laboratory of Automatic of Tlemcen, University of Abou bekr Belkaïd,PoBox 230, Tlemcen 13000, Algeria
a_hassam@mail.univ-tlemcen.dz

Z. Sari
Laboratory of Automatic of Tlemcen, University of Abou bekr Belkaïd,PoBox 230, Tlemcen 13000, Algeria
z_sari@mail.univ-tlemcen.dz

Abstract--**This paper presents the results of a simulation study of a typical flexible manufacturing system (FMS) that has routing flexibility. The objective in this study is to compare the following metaheuristics performances (Ant colony, genetic algorithms, simulated annealing, taboo search, particle swarm) with methods of selection of alternative routing in real time DMM (Dissimilarity Maximization Method) and modified DMM (modified Dissimilarity Maximization Method) in order to have an idea on the effectiveness from these metaheuristics and select the most effective. Results obtained after several simulations of this model FMS showed that all metaheuristics clearly improved the production rate, the utilisation ratio of the various machines and the utilisation ratio of the material handling system, for a saturated FMS and even in the presence of breakdowns.**

Index Terms-- Flexible Manufacturing Systems, Alternative routing, Routing selection rule, Metaheuristics, Simulation

1. Introduction

Flexible Manufacturing Systems (FMS) consist of a computer controlled and an integrated configuration of numerically controlled machine tools inter-linked with automated material handling systems. In a FMS each machine is quite versatile and capable of performing many different operations, therefore each part may have alternative routings in the system.

Scheduling involves decisions of allocating resources to tasks over time, and optimizing one or more objectives. Scheduling models can be either deterministic or stochastic. Deterministic models assume that all job data are known exactly in advance. In stochastic models, not all job data but their distributions are known. One of the earliest studies on the FMS scheduling problem in the work of Nof et al. [1] who demonstrated the importance of scheduling decisions for system performance.

From a traditional viewpoint, scheduling is an off-line activity where operations that are known prior to production are scheduled before the production starts but it can not react to internal or external perturbations. Because of this, rescheduling becomes obligatory in order to avoid the increase in the wait time, the work-in-process, and the weak use of equipments and consequently the degradation of the manufacturing system performances. Several researchers propose various methods to accommodate flexibility into off-line scheduling in order to increase the system performance [2]-[3]. However, real time scheduling has always remained a desirable but elusive goal [4]-[5].

The scheduling problems in manufacturing systems are generally NP hard and there are not universal methods making it possible to solve all the cases effectively [6].

Metaheuristics are the algorithms of the stochastic type aiming to solve a broad range of hard optimization problems, for which one does not know more effective traditional methods. Often inspired by analogies with reality like physics (simulated annealing, simulated diffusion,) biology (evolutionary algorithms, taboo search,) and ethology (ant colony, particle swarms …). They are generally of discrete origin, but can be adapted to the other types of problems and they share also the same disadvantages: difficulties of adjustment of the parameters of the method and the large computation time.

In this paper, our interest is focused to a group of metaheuristics, which include in particular the simulated annealing (SA), the genetic algorithm (GA), the taboo search (TS), the ant colony algorithms (ACO) and particle swarm (PSO); we are going to present a comparative study between these metaheuristics and DMM and modified DMM rules.

We used DMM and modified DMM methods as a comparison basis to evaluate the performances of the meta heuristics studied because it was shown in [7]-[8] that DMM and modified DMM methods present the best results for the alternative selection of routing when compared to other traditional rules.

The remainder of this paper is organized as follows. In the next section, we define DMM and modified DMM rules as well as metaheuristics which gave good results for our problem. In section 3, we define the FMS model. Section 4 is devoted to the results. Finally, conclusion and our perspectives are given.

2. SIMULATED METHODS AND METAHEURITICS

In this section, we are going to define DMM and modified DMM rules which are used for selecting an incoming part and later routing it to a machining centre for its next operation as well as the studied metaheuristics (Simulated annealing, Genetic algorithm, Taboo search, Ant colony algorithms and Particle swarm).

A. Dissimilarity Maximization Method

DMM [9] is developed with the goal of reducing the congestion in the system, The DMM concept is based on the objective of maximising the dissimilarities among the alternative routings. DMM uses a dissimilarity coefficient; which is based on the types of machines in routings. It selects a routing for each part so that the cumulative dissimilarity, in terms of machine tool requirements, is maximised. Dissimilarity between routings i and j is calculated by dividing the number of machine types that are not common in both routing i and j on the total number of machine types in both routings

B. Modified Dissimilarity Maximization Method [8]

This rule is also used in the selection of the alternative routings in real time in FMS. In DMM rule, after having selected a routing for a part, this routing cannot be used by another part as long as the first part did not leave the system thus each routing can contain only one part at the time. The modification of this rule, aims to keeping the same principle but by assigning several parts to only one routing. Then if all routings are selected, the following part will be transferred in the routing where the queue of the first machine of this routing, contains at least a free place.

C. Ant colony Optimization

This metaheuristic was introduced by Marco Dorigo (1992) [10] and was inspired by the studies on the real ant whose members are individually equipped with very limited faculties but can find the shortest path from a food source to their nest without visual cue. They are also capable of adapting to changes in the environment like the appearance of an unexpected obstacle on the initial path between the food source and the nest.

The first algorithm of this type of metaheuristics was conceived to solve travelling salesman problem. [10] This algorithm principle is simple.

When an ant k moves city i to city j, it leaves a trail on the way. Moreover, it chooses the next city to be visited using a probability P^{K}_{ij} based on a compromise between the intensity of the trail $\Gamma^k ij$ and visibility η_{ij} that represents the reciprocal of the distance between i and j, the relative importance of the two elements is controlled by two parameters α and β.

Each ant k has a form of memory $tabu_k$ it points out the ordered list of the cities which have been already visited in order to force this one to form an acceptable solution.

After a full run, each ant deposit a certain quantity of pheromone $\Delta\ \Gamma^k ij$ which depends on the quality of the solution found on the whole of its course.

This algorithm has been adapted to our problem by replacing the city i by the part i and the city j by the routing j. For each part i, the choice of routing j is based on a compromise between the intensity of the trail $\Gamma^k ij$ and visibility η_{ij} (depends on the number of parts in the input buffer of the first and second machine of the routing).

D. Simulated Annealing

The simulated annealing method was conceived by S.Kirkpatrick, C.D Gellat and M.P Vechi in 1983 [11]. It is a metaheuristic inspired by a process used in metallurgy to obtain a well ordered solid state with minimal energy called annealing process.

This technique consists in carrying material at high temperature, then to lower this temperature slowly.

This optimization method is based on works of N.Metropolis [12] which allow describing the behaviour of a system in thermodynamic equilibrium at a certain temperature. This technique transports the annealing process to the resolution of an optimization problem: the objective function to be minimized being the energy E of material. The temperature T is also introduced.

From an initial solution at a temperature T, we generate another solution close in a random way. If this solution improves the objective function, this latter is automatically accepted. If it degrades the function objective, it can also be accepted according to a probability $\exp(-\Delta E)$ were ΔE is the variation of the objective function, once thermodynamic equilibrium is reached one should lowers the front temperature slightly before implementing a new iteration.

E. Particle Swarms

PSO is a recent metaheuristic approach proposed by Kennedy and Eberhart in 1995[13].It is based on the metaphor of social interaction and communication, such as fish schooling and bird flocking when it is randomly searching for food in an area, where there is only one piece of food available and none of them knows where it is, but they can estimate how far it would be at each iteration. For this problem, the simplest strategy to find and get the food is to follow the bird known as the nearest one to the food.

In PSO, each single solution is called a particle, the group becomes a swarm (population) and the search space is the area to explore. Each particle has a fitness value calculated by a fitness function, and a velocity of flying towards the optimum. In the original version of PSO, all particles fly across the problem space following the particle nearest to the optimum by two elastic forces. One attracts it to the best location so far encountered by the particle. The other attracts it with random magnitude to the best location encountered by any member of the swarm.

Each particle maintains two character items: velocity and position. Both of them are updated at each step until the population converges to an optimum as follow:

$$V_i(t) = V_i(t-1) + c_1 r_1 (p_i + x_i(t-1)) + c_2 r_2 (p_g + x_i(t-1)) \quad (1)$$

$$X_i(t) = X_i(t-1) + V_i(t) \quad (2)$$

Where v_i (t) denotes the velocity vector of particle j at time t. x_i (t) represents the position vector of particle j at time t. Vector p_i is the memory of particle i at current generation, and vector p_g is the best location found by the whole swarm. Cognitive coefficient c_1 and social coefficient c_2 are known as acceleration coefficients r_1 and r_2 are two random numbers with uniform distribution.

PSO originally designed for continuous optimization problems, but can be adapted to discrete problems like our problem of routing selection where (1) and (2) were replaced by the following equation proposed by Pan et al [14]:

$$X_i(t) = c_2 \oplus F_3(c_1 \oplus F_2(w \oplus F_1(X_i(t-1)), p_i(t-1)), G(t-1)) \quad (3)$$

The equation (3) consists of three components: The first component is $\lambda_i(t) = w \oplus F_1(X_i(t-1))$ which represents the velocity of the particle. F_1 represents an operator which modifies the routing of some parts with the probability of w, a uniform random number r is generated between 0 and 1. If r is less than w then the F_1 is applied to generate a perturbed permutation of the particle by $\lambda_i(t) = F_1(X_i(t-1))$, otherwise current permutation is kept as $\lambda_i(t) = X_i(t-1)$.Of the same way, the second component which is cognition part of the particle $\delta_i(t) = c_1 \oplus F_2(\gamma_i(t), p_i(t-1))$ and the third component which is the social part of the particle $X_i(t) = c_2 \oplus F_3(\delta_i(t), G_i(t-1))$ have been modified where F_2 and F_3 represent the crossover with the probability C_1 and C_3.

F. Genetic algorithms

Genetic algorithms were proposed by Holland [15]. They were inspired from the principles of natural genetics and the theory of evolution (The presence or absence of genes and their order in the chromosome decide the characteristics of a species. Different traits are passed on from one generation to the next through different biological processes that operate on the genetic structure………..).

In a GA, each solution is stored in an artificial chromosome represented by a code. Each of these chromosomes is defined by two characteristics.

The first is their genotype, which is the actual sequence which defines the chromosome. It is called like this because of the analogy with a genetic sequence in biology. The second is the phenotype, which is the decoded version of the genotype that determines the traits of the individual.

With each of the chromosomes, the parameters are decoded and evaluated by the fitness function to determine the quality of the phenotype.

New candidates are generated gradually from a set of renewed populations by applying artificial genetic operators selected, after repeatedly using operators of crossover and mutation [16].

Crossover is performed by taking two fit genotypes, choosing a place along the bit string, cutting each of them at that place and then connecting one string's left to the other string right and vice versa. This produces two new chromosomes, which are a combination of the two parents.

Reproduction is simply a matter of passing chromosomes which are judged to be above a certain fitness level through to the next generation and mutation is done by choosing bits randomly and swapping them.

G. Taboo search

This method was formalized by F Glover [17]. It is based on the use of mechanisms inspired by the human memory. The principle of this metaheuristic is simple: we generate an initial configuration which is updated during successive iterations. The mechanism of passage of one configuration, called *s*, to the next one, called *t*, comprises two stages:

- The first builds the set of the neighbours of s, i.e. the set of the accessible configurations in only one elementary movement of s, let V (s) be the set (or the subset) of these neighbours.
- The second evaluates the objective function *f* of the problem for each configuration belonging to *V* (*s*). The configuration *t*, which succeeds *s* in the series of the solutions built by the taboo method, is the configuration of *V* (*s*) in which f takes the minimal value. This configuration *t* is adopted even if it is worse than *s*; due to this characteristic the taboo method can avoid the trapping in the local minima.

To avoid to return to a retained configuration and generate a cycle in each iteration the taboo list that gave its name to the method contains *m* movements ($t \rightarrow s$), which are the opposite of the last *m* movements ($s \rightarrow t$) carried out.

4. Flexible Manufacturing System Model Presentation

The FMS considered (see fig.1) for this study includes seven machining centres, each machine has an input and output buffer, a loading and an unloading station, and six different part types, each part types has different alternative routings.

The alternative routes and processing times of each part type and the production ratio of the part types that are randomly arriving at the loading station are shown in Table I (see table I)

The operation of the FMS model used in this study is based on the following assumptions:

- The flexible process plan of each part type is known prior to production.
- Processing times are known deterministically and they include tool change, set-up, and machining times.
- The processing time of an operation is the same on the alternative machines identified for that operation.
- Each machine can process only one part at a time.

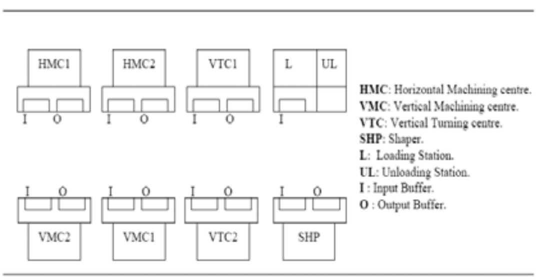

Fig. 1.Configuration of the FMS model [5]

5. Results And Discussions

To validate the results of this study and show the improvements made by the metaheuristics we simulated DMM and modified DMM on a FMS model using ARENA software (1). On the other hand, metaheuristics were simulated using Java (2) on the same model with variations on the criteria of the studied system.

The methods and the algorithms were simulated over 20000 hours with a warm up time of 3000 hours. Simulations has been carried out on (Core (TM) 2Duo CPU with 2.2 GHZ and 1 GO of RAM)

In this section we will have some results obtained after the simulation of the two methods and execution of the programs as well as interpretations of these results.

A. Production rate

Rate of part leaving the system is calculated by dividing the number of parts left by the number of parts created.

TABLE I

Alternative Routings of Part Types [5]

Part type and Production RATIO	Routing and processing time (min)
A 17%	L – VTC1 (30) – VMC1 (20) - UL
	L – VTC1 (30) – VMC2 (20) - UL
	L – VTC2 (30) – VMC1 (20) - UL
	L – VTC2 (30) – VMC2 (20) - UL
B 17%	L – VTC1 (20) – SHP (1) – VMC1 (15)-UL
	L – VTC1 (20) – SHP (1) – VMC2 (15)- UL
	L – VTC2 (20) – SHP (1) – VMC1 (15) - UL
	L – VTC2 (20) – SHP (1) – VMC2 (15) - UL
C 17%	L – VTC1 (40) – VMC1 (25) - UL
	L – VTC1 (40) – VMC2 (25) - UL
	L – VTC2 (40) – VMC1 (25) - UL
	L – VTC2 (40) – VMC2 (25) - UL
D 21%	L – VTC1 (40) – SHP (1) – VTC1 (20) – HMC1 (35)–UL
	L – VTC1 (40) – SHP (1) – VTC1 (20) – HMC2 (35)–UL
	L – VTC1 (40) – SHP (1) – VTC2 (20) – HMC1 (35)–UL
	L – VTC1 (40) – SHP (1) – VTC2 (20) – HMC2 (35)–UL
	L – VTC2 (40) – SHP (1) – VTC1 (20) – HMC1 (35)–UL
	L – VTC2 (40) – SHP (1) – VTC1 (20) – HMC2 (35)–UL
	L – VTC2 (40) – SHP (1) – VTC2 (20) – HMC1 (35)– UL
	L – VTC2 (40) – SHP (1) – VTC2 (20) – HMC2 (35)–UL
E 20%	L – VTC1 (25) – SHP (1) – VTC1 (35) – HMC1 (50)–UL
	L – VTC1 (25) – SHP (1) – VTC1 (35) – HMC2 (50)–UL
	L – VTC1 (25) – SHP (1) – VTC2 (35) – HMC1 (50)–UL
	L – VTC1 (25) – SHP (1) – VTC2 (35) – HMC2 (50)–UL
	L – VTC2 (25) – SHP (1) – VTC1 (35) – HMC1 (50)–UL
	L – VTC2 (25) – SHP (1) – VTC1 (35) – HMC2 (50)–UL
	L – VTC2 (25) – SHP (1) – VTC2 (35) – HMC1 (50)–UL
	L – VTC2 (25) – SHP (1) – VTC2 (35) – HMC2 (50)–UL
F 8%	L –HMC1 (40) – UL
	L –HMC2 (40) – UL

The figure 2 and table II show that for a significant rate of creation of the parts results obtained by metaheuristics are better than those of the Modified DMM (MDMM) and DMM and that below the creation rate of 1/25 the production rate is practically the same one for all methods.

On the other hand, the rates obtained by simulated annealing and particle swarm are better than that obtained by other metaheuristics.

TABLE II

RATE OF PART LEAVING THE SYSTEM FOR QUEUE SIZE=2.

creation rate	1/5	1/10	1/15	1/20	1/25	1/30	1/35	1/40
SA	24.3	48.7	73.1	97.7	99.9	99.9	99.9	99.9
PSO	24.3	48.8	73.0	97.5	99.9	99.9	99.9	99.9
ACO	24.0	48.1	72.1	96.1	99.9	99.9	99.9	99.9
TS	23.8	47.6	71.5	95.2	99.9	99.9	99.9	99.9
GA	22.3	45.0	67.5	90.0	99.9	99.9	99.9	99.9
MDMM	21.1	41.6	60.7	84.4	99.7	99.9	99.9	99.9
DMM	8.8	15.4	32.0	24.6	81.4	99.9	99.9	99.9

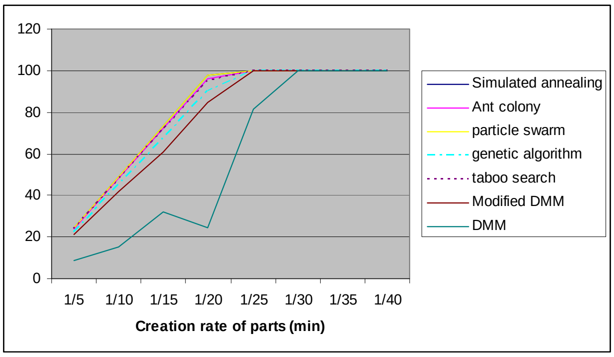

Fig. 2.Rate of part leaving the system for queue size=2

B.Utilisation rate of the machines

The utilization rate of machines is a very significant criterion in the measurement of the performance of a production system. The utilization rate for the machines VTC 1 and VTC2 is larger for the ant colony and particle swarms than for the modified DMM and DMM and other metaheuristics when rate of creation increase (see figure 3 [1]or table III).The most efficient method is ant colony.

TABLE III

UTILIZATION RATE OF THE MACHINES VTC1 AND VTC2 FOR QUEUE SIZE=2.

creation rate	1/5	1/10	1/15	1/20	1/25	1/30	1/35	1/40
SA	92.6	92.5	92.9	92.9	78.2	65.7	56.6	49.8
PSO	93.7	93.7	93.6	93.7	78.6	65.2	56.5	49.6
ACO	94.7	94.7	94.6	94.6	78.7	65.6	56.2	49.2
TS	91.8	91.7	91.8	91.5	78.7	65.6	56.2	49.2
GA	87.1	87.7	87.6	87.4	79.0	66.0	56.6	49.3
MDMM	84.7	83.5	82.7	84.5	79.8	66.5	56.9	49.9
DMM	35.5	30.9	42.5	24.6	65.0	66.5	57	49.9

(1): Java is a programming language developed by Sun Microsystems
(2): ARENA is simulation software marked by Rockwell Automation

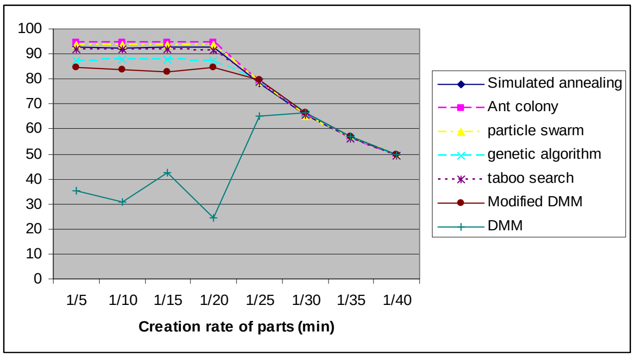

Fig. 3.Utilization rate of the machines VTC1 and VTC2 for queue size=2.

C. Utilisation rate of the handling system

Figure 4 and table IV show us that for a saturated system the auto guided vehicle (AGV) utilization rate is larger for particle swarm and ant colony than the modified DMM and DMM and other metaheuristics, which is due to the high production rate and the increase in the use of the machines.

The best results concerning the rate of handling system are obtained by the ant colony.

TABLE IV

UTILIZATION RATE OF AGV FOR QUEUE SIZE=2.

creation rate	1/5	1/10	1/15	1/20	1/25	1/30	1/35	1/40
SA	33.2	33.1	33.1	33.2	28.3	23.8	20.6	18.1
PSO	33.7	33.7	33.6	33.7	28.5	23.6	20.5	18
ACO	34.3	34.3	34.3	34.3	28.5	23.8	20.4	17.8
TS	33	33	33	32.9	28.5	23.8	20.4	17.8
GA	31.4	31.6	31.6	31.5	28.7	24	20.5	17.9
MDMM	30.4	30.1	29.1	30.2	27.3	21	17.4	14.9
DMM	12	10.5	14.3	8.4	21.6	20.9	17.4	15

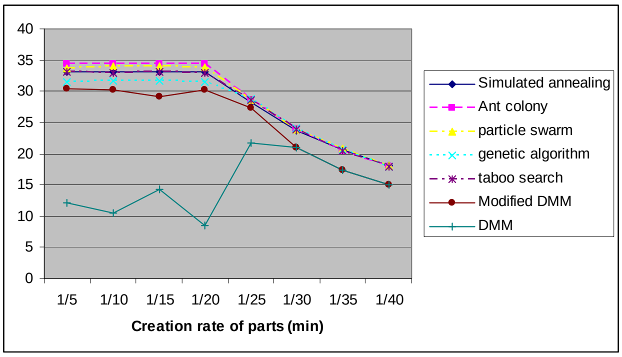

Fig.4.Utilization rate of the handling system for queue size=2.

6. CONCLUSION

In this paper we have presented certain metaheuristics and compared their performances with methods of selection of alternative routing in real time DMM (Dissimilarity Maximization Method) and modified DMM.

Results obtained showed that all metaheuristics gave results better than DMM and modified DMM and clearly increased the performances of the system for a saturated production system and high rate of creation of the parts because they increase the production rate and the utilization rated of machines and the utilization of AGV.

Results showed that simulated annealing gives the best results concerning the production rate and the ant colonies are more efficient if one is interested in utilization rate of the machines and of AGV so there is not a metaheuristic which is better than all methods and metaheuristics for all performances. We have to notice also that these techniques could not improve the performances concerning the cycle time and the rate of work in progress of the system.

Our future work is to combine several metaheuristics to improve performances of the system and found results better than these presented.

REFERENCES

[1] S. Nof, M. Barash and J. Solberg. "Operational control of item flow in versatile manufacturing system", International journal of production research, 17, 479-489, 1979.

[2] C. Saygin and S.E. Kilic. "Integrating flexible manufacturing systems with scheduling in flexible manufacturing system", International journal of advanced Manufacturing Technology, 15(4),268-280, 1999.

[3] C. Saygin and S. E. Kilick. "Effect of flexible process plans on performance of flexible manufacturing systems", proceedings of 7th International DAAM symposium, Vienna, Austria, 393-394, 1996.

[4] C. Basnet and J. H. Mize. "Scheduling and control of flexible manufacturing systems: a critical review", International journal of Computer Integrated Manufacturing, 7(6), 340-355, 1994.

[5] C. S. Shukla and F. F. Chen. "The state of the art in intelligent real-time FMS control: a comprehensive survey", Journal of intelligent Manufacturing, 7, 441-455, 1996.

[6] M.R Garey and D.S. Johson.,"Computers and intractability a guide of the theory of NP- completeness", W.H. Freeman and company, San Fransisc, 1979.

[7] L. GHOMRI et Z. SARI, " Influence des Contraintes et des Perturbations sur les Performances des Règles de Routage dans un FMS ", Rabat, Conception et Production Intégrées, 2007.

[8] A. HASSAM and Z.SARI," Real-Time Selection of Process Plan in Flexible Manufacturing Systems: Simulation Study", CHINA, International Conference on Industrial engineering and Systems Management, 2007.

[9] C. Saygin, F.F. Chen and J. Singh, "Real-Time Manipulation of alternative Routings in Flexible Manufacturing Systems: A simulation Study", International journal of advanced Manufacturing Technology, 18, 755-763, 2001

[10] M. Dorigo, "Optimization, learning and natural algorithms", Italy, PHD thesis, DEI, Politecnico di Milano, 1992.

[11] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi. "Optimization by Simulated Annealing", Science Number 4598, volume 220, 4598, p .671-680, 1983.

[12] N.Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller. "Equations of state calculations by fast computing machines", Journal of Chemical Physics, 21:1087–1091, 1953.

[13] R. C. Eberhart and J. Kennedy, " New optimizer using particle swarm theory", Proceedings of the 6th International Symposium on Micro Machine and Human Science, pp. 39–43, Nagoya, Japan, October 1995.

[14] Pan QK, Tasgetiren MF, Liang YC, "A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem with makespan criterion", Proceedings of the international workshop on UK planning and scheduling special interest group, UK PLANSIG2005. City University, London, pp: 31–41, 2005.

[15] Holland, J.H., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor (1975).

[16] Goldberg, E.E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA (1989).

[17] Glover F. and Manual Laguana, 1997. *Tabu Search*. Article was principally adapted from the book « Tabu Search » of Glover, Laguana 1997, Kluwer Academics publishers.